# BOOKS RECOMMENDATION SYSTEM

Project report

By

Saynur Altın



## Department Of Computer Engineer



## Aydın Adnan Menderes University

# ABSTRACT

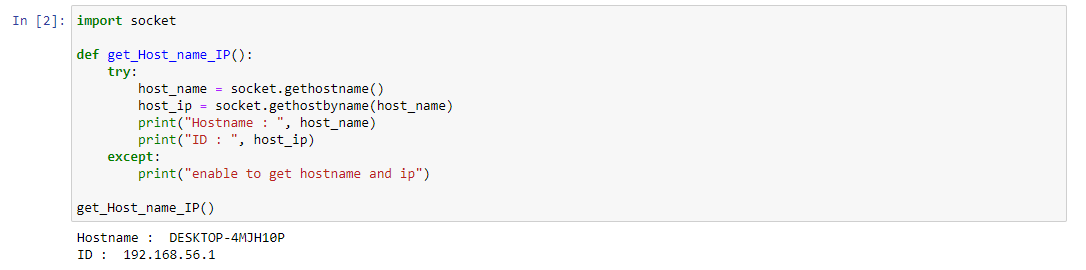


This is “**BIG DATA ANALYSIS FINAL PROJECT** ” submitted by Saynur Altın in fulfillment for the final project of Big Data Analysis. This work is original and has not been submitted partially or fully anywhere else for any other degree or diploma. This project was an attempt at developing a recommendation system for books using Spark mllib. I built a recommendation engine using collaborative filtering and I used ALS (Alternating Least Squares) for training recommendation model. The aim of this project built a book recommender system based on collaborative ﬁltering using Apache Spark.

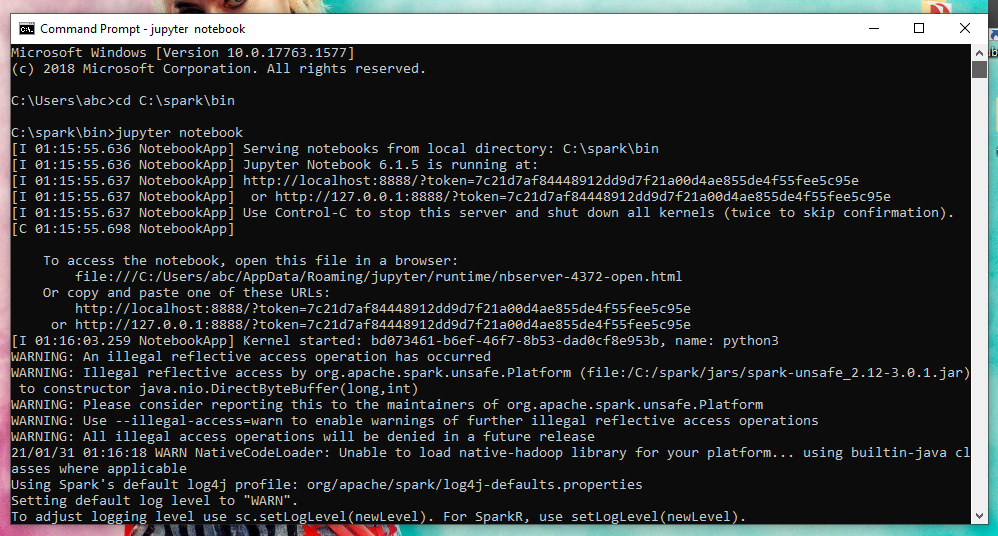
# TABLE OF CONTENTS

1. COMPUTER INFO
2. DATA INFORMATION AND DATA PREPROCESSING
3. ALS MODEL
4. RECOMMENDATION
5. EXPERIMENTAL RESULTS ANALYSIS AND DISCUSSION
6. WARNING AND SOLUTION
7. **COMPUTER INFO**

**In python code**

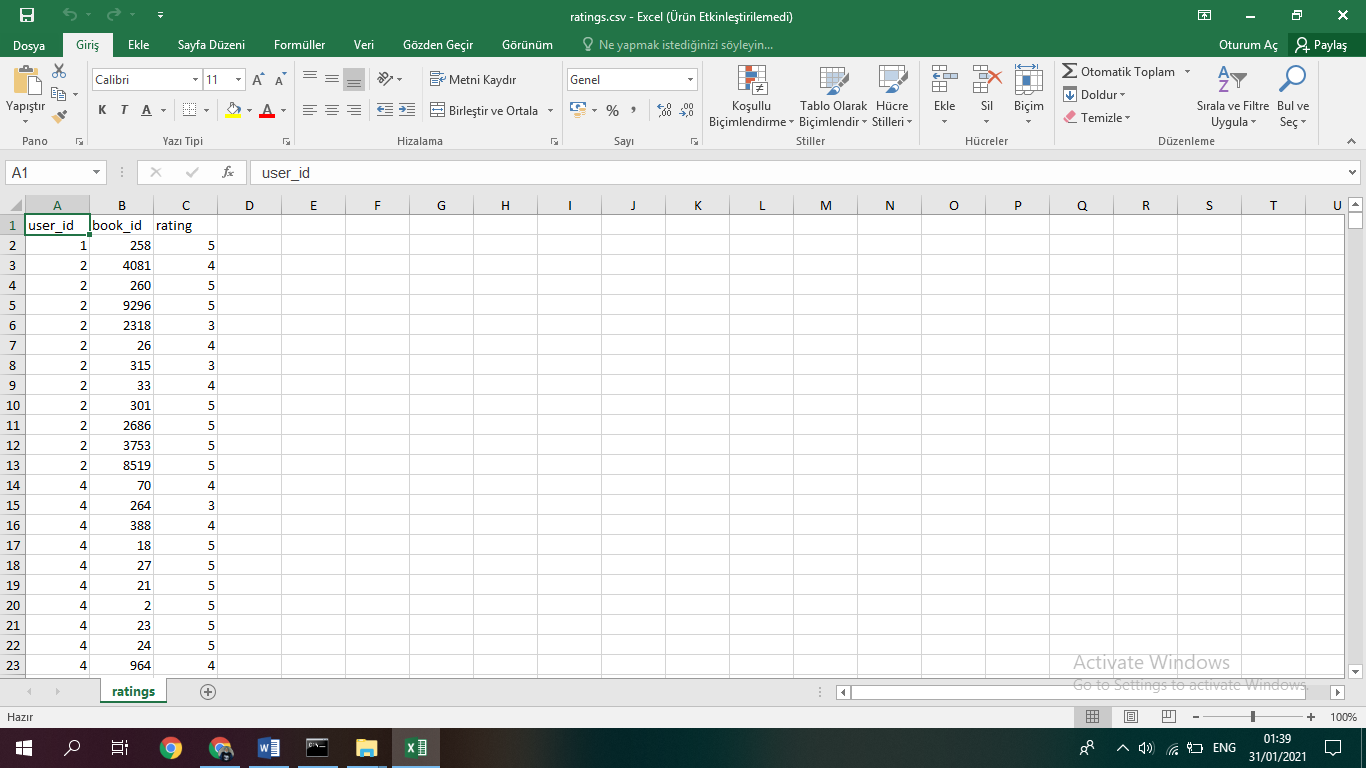
****

**In CMD**

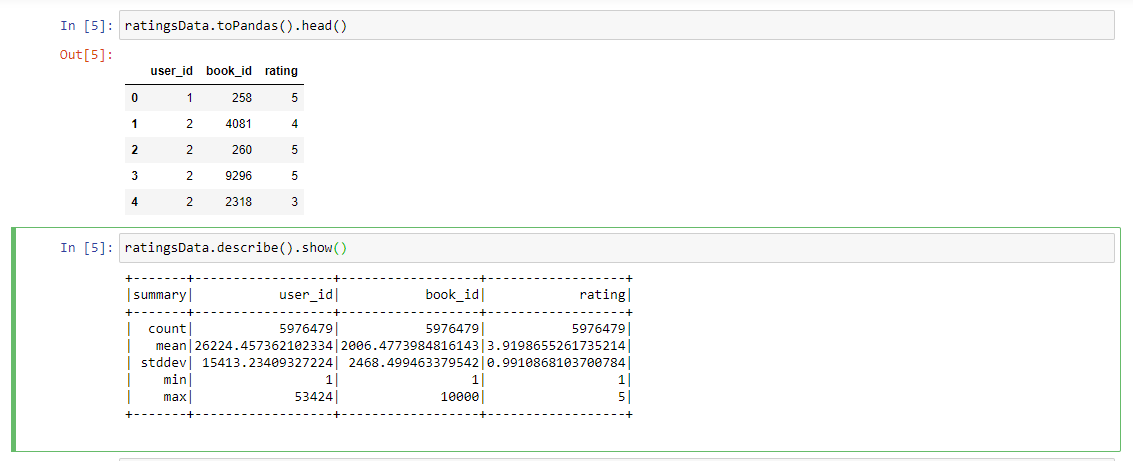
****

1. **DATA INFORMATION AND DATA PREPROCESSING**

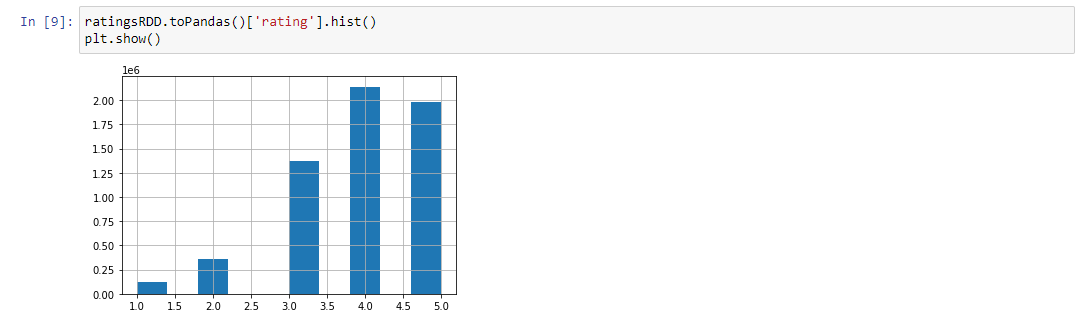
**RATINGS.CSV**



In most cases, there are at least 10 books rated by each user.

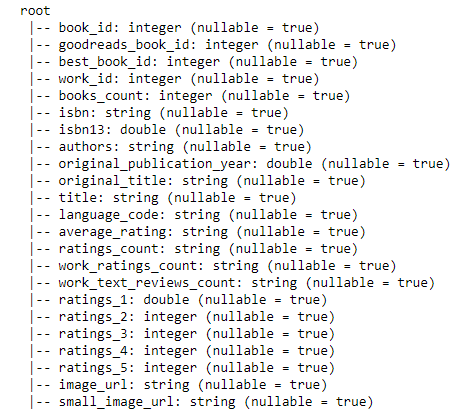


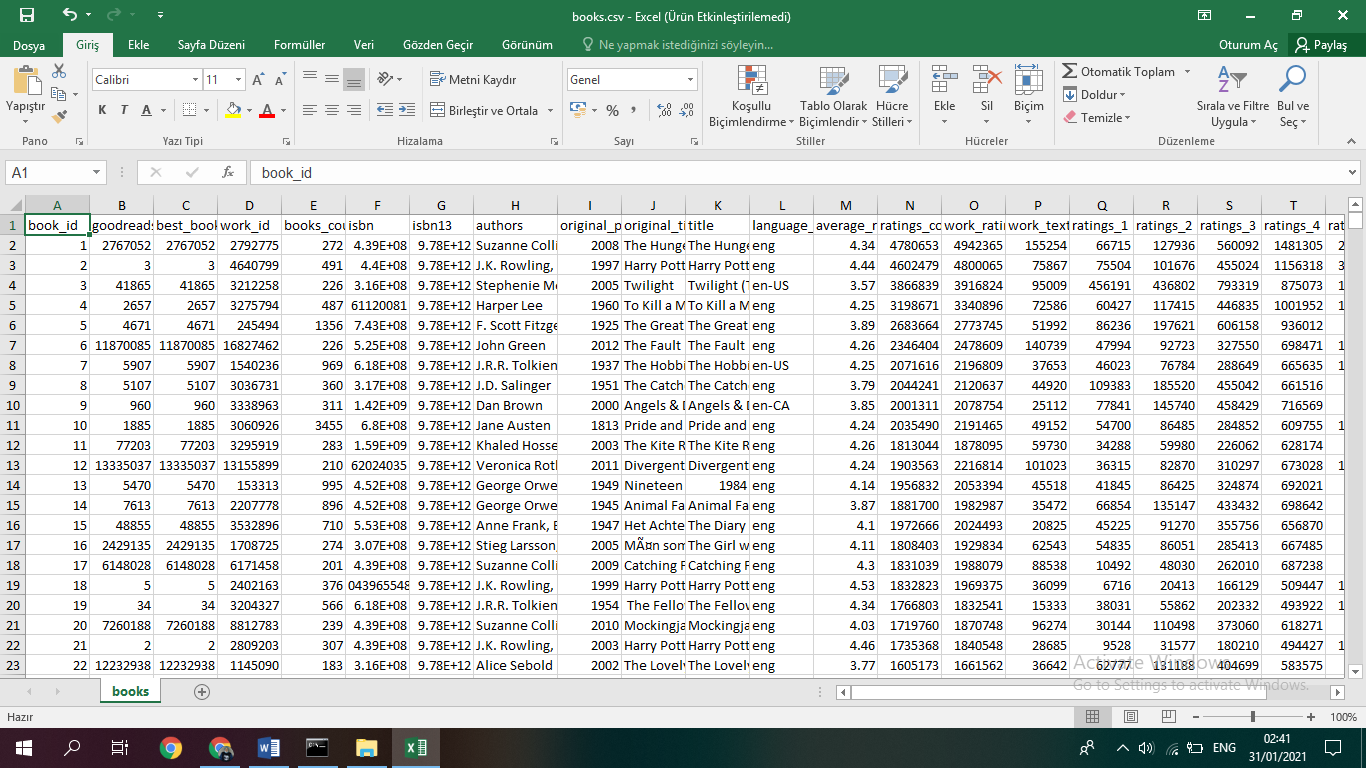
The ratings.csv file contain user\_id, book\_id and rating. Ratings dataset contains about 6 million ratings across **10000 different books**. There are about 50000 users and the rating between 1 and 5. I will use these about 6 million ratings to predict ratings for all the other books that haven’t been read by the users. Later, I will combine ratings.csv and books.csv according to book\_id to offer books suggestions to our users (because book\_id is identic data and if I will combine ratings.csv and books.csv according to book\_id, my suggestions would be more meaningful).



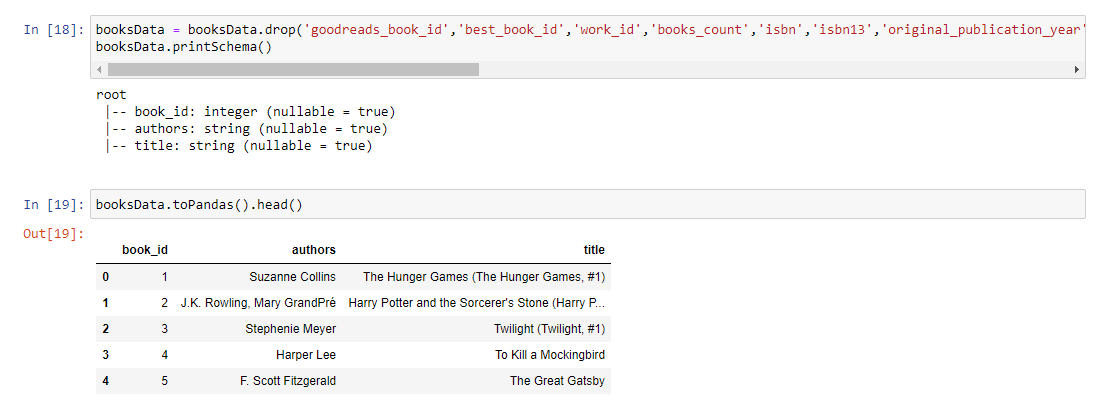
Ratings plot in ratings.csv

**BOOKS.CSV**

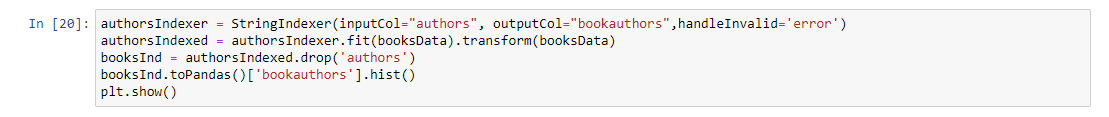




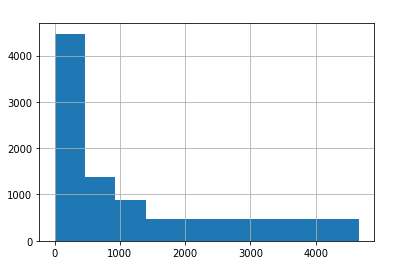
I don’t need all columns. I need only book\_id, tittle and authors columns so I drop others columns in data and create new data

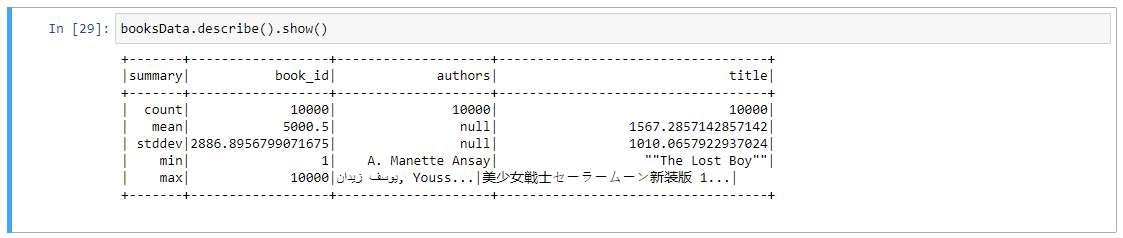


New data has only book\_id, authors and title columns. Thus, I got rid of the data crowd. So that, I can make analysis and recommendation easier.



I indexed String Author value to visualize authors column.





There are 10,000 books in books.csv

1. **ALS MODEL**

I used the ALS algortithm in Spark ML Library to build a model and predict the recommendations for the user.



I splited the ratings data into two portions which are training and test data to parse their lines once they are loaded into RDDs. Parsing the books and rating ﬁles yields two RDDs: For each row in the ratings dataset, There are user\_id, book\_id and rating. In order to determine the best ALS parameters for our experiments, we need to break up the ratings RDD dataset into two pieces as follows: a training set which I used 70% of the data to train models and test set, which used 30% . And seed is 5011.



Here’s a breakdown of the parameters used when creating the ALS object:

* maxIter : Maximum number of iterations to run (default: 10)
* regParam : Regularization parameter. Reduces overfitting your model which leads to the reduction of variance in estimates; however, it comes with the expense of adding more bias. (default: 0.01)
* rank : Size of the feature vectors to use. Larger ranks can lead to better models but are more expensive to compute (default: 10)
* userCol : Name of the column containing user IDs (default: user)
* itemCol : Name of the column containing item IDs (default: item)
* ratingCol : Name of the column containing the ratings (default: rating)

I used book\_id for itemCol

I used 0.01 and 0.1 regParams with 5, 8, 10 iterations for 1, 2, 3 ranks. So That, I created a total of 18 models that will be tested and tuned before we receive the final model.

I used ratings\_train data to train als model(model = als.fit(ratings\_train))

I have a trained model. I made predictions with use trained model and ratings\_test data.( predictions = model.transform(ratings\_test))

I used Evalator from pyspark ml library to see the accuracy of the trained data with Regression Evaluator.( new\_predictions = predictions.filter(col('prediction') != np.nan)

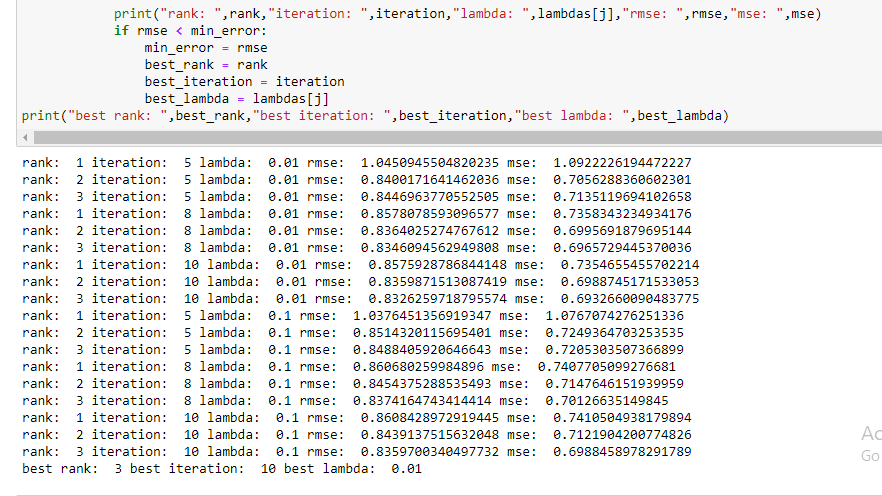
evaluator = RegressionEvaluator(metricName="rmse", labelCol="rating", predictionCol="prediction")

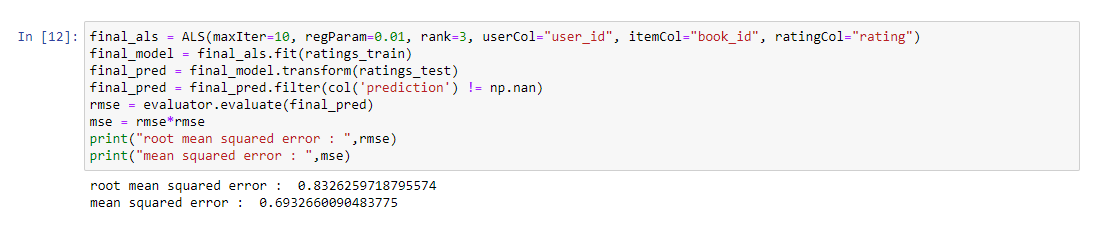
rmse = evaluator.evaluate(new\_predictions)

mse = rmse\*rmse

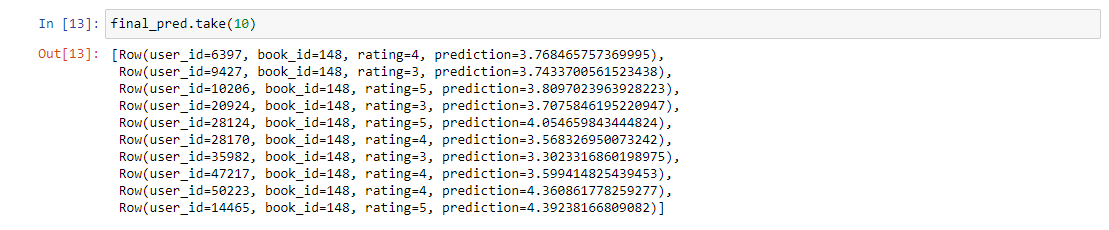
errors.append(rmse))

Finally I printed 18 models and choose best model.



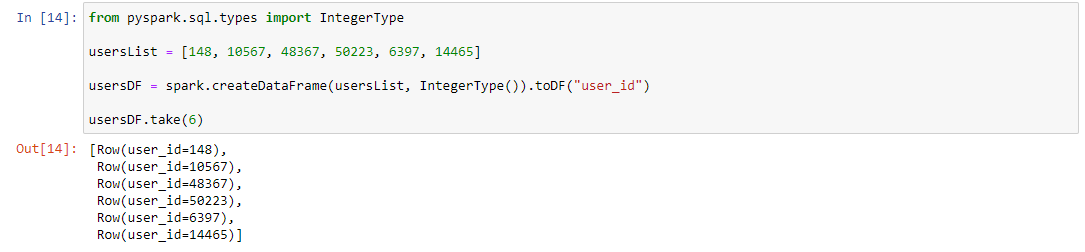


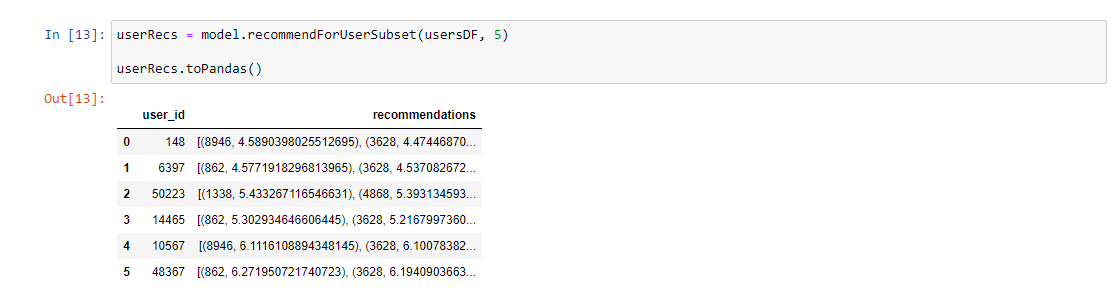
I built best Model and made final predictions using best rank best iteration and best data.



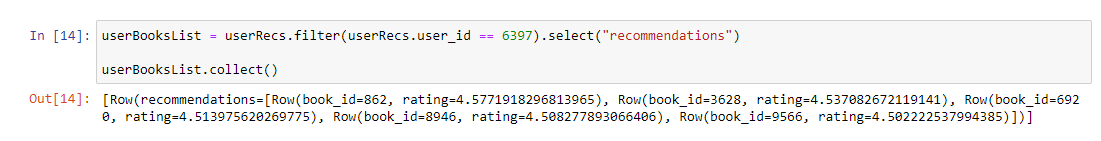
The prediction on the test set shows that it is very close to the original rating. Like for example the rating for user\_id 50223 was originally 4 and our algorithm predicted it to be 4.3 which is pretty close.

1. **RECOMMENDATION**

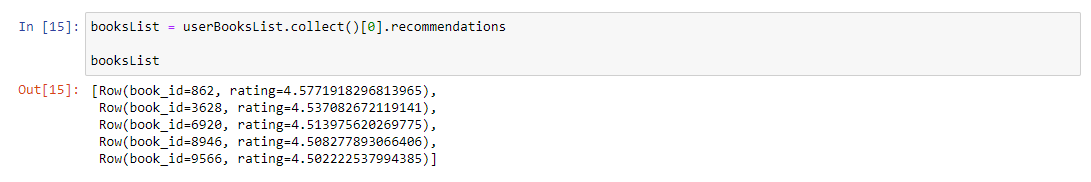
****



I choose 5 users and created users list and convert data frame this user list to make recommendation for these 5 users.



I suggest books for only 6397 user\_id.



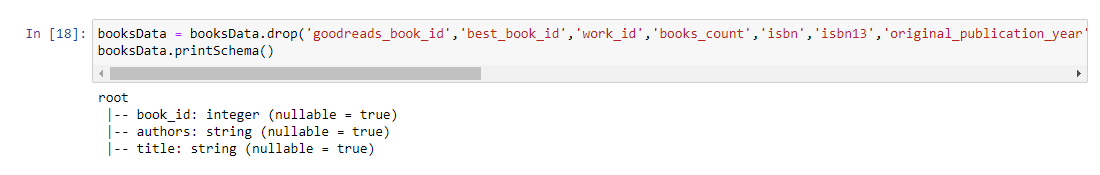
I bought and printed the recommended books for the 6397 user\_id



I created booksDF data frame from recommended books for the 6397 user\_id.

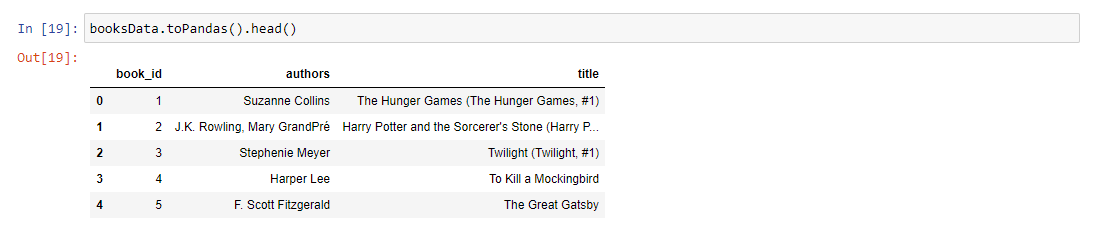


I read books.csv and create books Data

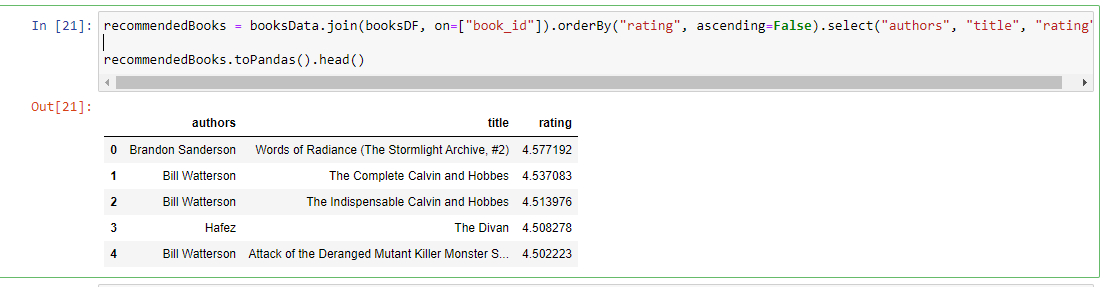


There are too much column that I won't use so I drop column that I won’t use.

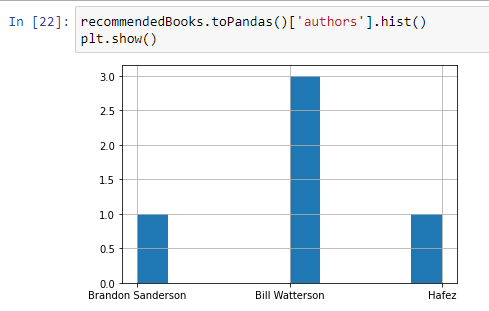
I only used book\_id, authors and title columns.

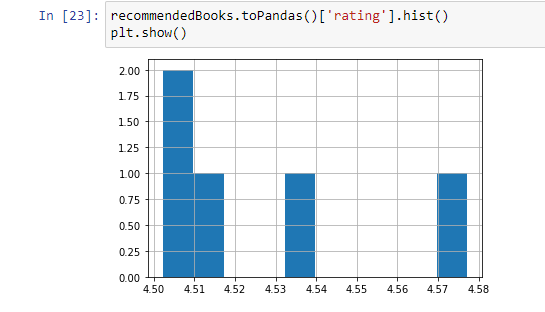


First 5 books data

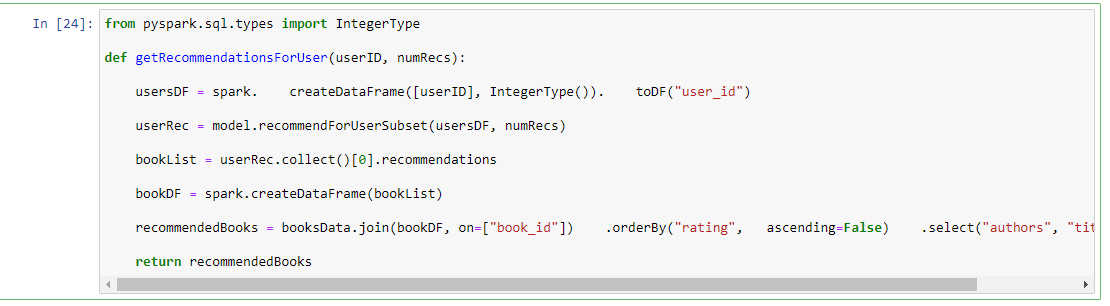


I made recommendation with book\_id on ratings data and book\_id on books data for the 6397 user\_id from booksDF.





Histograms for recommended books for the 6397 user\_id.



I created Recommendation function for users like above recommendation for the 6397 user\_id. But this function making recommendation for all users.

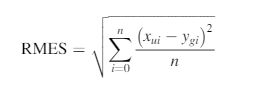
For example 10 and 5 recommended books for the user I specified.





1. **EXPERIMENTAL RESULTS ANALYSIS AND DISCUSSION**

In the proposed model, root mean squared error (RMSE) is used as a performance measure. RMSE works by measuring the difference between error rate a user gives to the system and the predicted error by the model. Equation depicts how RMSE works on book recommender system.

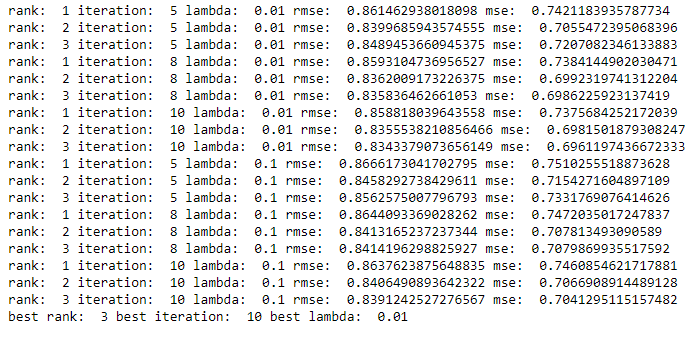


Where by *Xui* is the rating that user u gives to an item *i* in the experimental data, *Ygi* is a predicted rating that the movie that user *u* gives to an item and where *n* is the number of ratings in the test data.

In this project, ApacheSpark is used to demonstrate an efﬁcient parallel implementation of a collaborative ﬁltering method using ALS. ALS is used for dimensionality reduction purpose which helps in overcoming the limitations of collaborative ﬁltering such as data sparsity and scalability. The challenges of data sparsity are appearing in numerous situations, speciﬁcally, another problem, when a new an item or user has just added to the system, it is difﬁcult to ﬁnd similar ones since there is no sufﬁcient information, this problem is called cold start problem. When selecting the ALS algorithm as a part of building the proposed movie recommender system, there is basic parameter through them can determine the best rating of users for given books. These parameters are Rank, Iterations, and Lambda.

The aim of this section is to study and determine the selection of parameters that affect the performance of ALS model in building a book recommender system. The parameters, lambda, and iterations are used in order to control and adjust the predicting capability of matrix factorization which is depending on ALS technique which in turn affect the evaluation of movie RS. The iterations and lambda parameters are used as follows: Lambda which speciﬁes the regularization parameter in ALS and iterations in which the proposed model should run the speciﬁed number of iterations. The ALS algorithm achieves its optimal solution between 5 and 20 iterations. The following codes show the performance of book recommendation engine based on ALS under different values of lambda and iteration.





This output indicates the RMSE with the changes of lambda with iterations parameters and ranks of ALS model. Using these three metrics, 18 models are achieved as shown outputs.

In general, the lowest value of the RMSE is considered the best case for prediction in building recommendation system. Therefore, we will adopt the 9th model because the value of the RMSE is smaller.

1. **WARNING AND SOLUTION**

**Py4JJavaError**: An error occurred while calling o702.fit.

: org.apache.spark.SparkException: Job aborted due to stage failure: Task 2 in stage 746.0 failed 1 times, most recent failure: Lost task 2.0 in stage 746.0 (TID 5331, DESKTOP-4MJH10P, executor driver): java.lang.StackOverflowError

I GET THIS ERROR FOR ITERARION VALUES. WHEN I USE (50,100,200) ITERATIONS AND (50,100,200) RANKS VALUES I GET AN ERROR SO I CHANGE THIS VALUES. I USED (5,8,10) FOR ITERATIONS AND (1,2,3) FOR RANKS.